

APPLICATION FOR UNITED STATES LETTERS PATENT

For

IMPROVING STORE PERFORMANCE

Inventor(s):

Vladimir Pentkovski
Ling Cen
Vivek Garg
Deep Buch
David Zhao

Prepared by:

BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP
12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1030
(408) 720-8300

"Express Mail" mailing label number: EV409356210US

Date of Deposit: March 30, 2004

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to Mail Stop Patent Application, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450

Anne Collette

(Typed or printed name of person mailing paper or fee)

Anne Collette

(Signature of person mailing paper or fee)

3/30/2004

(Date signed)

IMPROVING STORE PERFORMANCE

FIELD

[0001] Embodiments of the invention relate to microprocessor architecture. More particularly, embodiments of the invention relate to a method and apparatus to improve store performance in a microprocessor by allowing out-of-order issuance of read-for-ownership operations and more efficiently using the store buffer latency periods.

BACKGROUND

[0002] A microprocessor typically communicates with a computer system via a shared computer system bus known as a “front-side bus” (FSB). However, as microprocessor performance is improved and as computer systems use multiple processors interconnected along the same FSB, the FSB has become a performance bottleneck.

[0003] One approach to this problem is the use of point-to-point (PtP) links between the various processors in a multiple processor system. PtP links are typically implemented as dedicated bus traces for each processor within the multi-processor network. Although typical PtP links provide more throughput than FSB, the latency of PtP links can be worse than the latency of FSB.

[0004] Latency of PtP can particularly impact the performance of store operations performed by microprocessor, especially in microprocessor architectures requiring strong ordering among the store operations. Because of the strong ordering requirements, for example, previously issued store operations must typically be accessible, or at least detectable, to other bus agents within the system before later

store operations may be issued by the processor. The detectability of an operation, such as a store, load, or other operation, to other bus agents within a computer system is often referred to as “global observation” of the operation. Typically, microprocessor operations or instructions only become globally observable after they have been stored to a cache or other memory in which other agents in the system may detect the presence of the operation or instruction.

[0005] In the case of store operations within a strong ordering microprocessor architecture, typical microprocessors will not issue a store operation from a store buffer, or other store queuing structure, or, in some cases, from the processor execution unit, until the previous store operation has been globally observed. The issuance of a store operation in typical microprocessor architectures is preceded by an operation, such as read-for-ownership (RFO) operation, to gain exclusive control of a line of the cache or other storage area in which the store operation is to be stored so that it may be globally observed. However, in typical microprocessor architectures, RFO operations are not issued until preceding store operations are globally observed.

[0006] Figure 1 illustrates a prior art cache architecture for handling issued store operations within a strongly ordered microprocessor architecture. The store buffer contains data X_1 and Y_1 that are to be stored in addresses X and Y , respectively of the level-1 (L1) cache via the cache line fill buffer (LFB). However, in typical prior art architectures, neither the store data, X_1 and Y_1 , nor their corresponding RFO operations may be issued until the data X_0 and address X in the L1 cache has been globally observed.

[0007] Due to latency in the issuance, and ultimately the retiring, of store operations within prior art architectures, the overall performance of a microprocessor and the system in which it exists may be compromised. Furthermore, as PtP multiple processor systems become more pervasive, the problem may be exacerbated as each processor in the system may dependent upon data being stored by other processors within the system.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] Embodiments of the invention are illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like references indicate similar elements and in which:

[0009] Figure 1 illustrates a prior art cache architecture for handling issued store operations within a strongly ordered microprocessor architecture.

[0010] Figure 2 illustrates a computer system in which at least one embodiment of the invention may be used.

[0011] Figure 3 illustrates a bus agent in which at least one embodiment of the invention may be used.

[0012] Figure 4 illustrates one embodiment of the invention in which a global observation store buffer (GoSB) is used to track store operations and store corresponding data values that have become globally observable.

[0013] Figure 5 illustrates an embodiment of the invention in which the GoSB index and GoSB valid fields are not stored within level-1 (L1) cache or line-fill buffer (LFB) entries, but instead, the GoSB index field is stored within entries of the store buffer.

[0014] Figure 6 is a flow chart illustrating operations associated with at least one embodiment of the invention.

DETAILED DESCRIPTION

[0015] Embodiments of the invention relate to microprocessor architecture. More particularly, embodiments of the invention relate to a method and apparatus to improve store performance in a microprocessor by allowing out-of-order issuance of read-for-ownership (RFO) operations and more efficiently using the store buffer latency periods.

[0016] In order to facilitate out-of-order RFO operations while improving store buffer efficiency, at least one embodiment of the invention involves using a storage medium, such as a globally observable store buffer (GoSB), to keep track of store data that has become globally observable. Tracking globally observed data within the GoSB allows store data to be stored within snoop'able storage devices, such as a level-1 (L1) cache and a line-fill buffer (LFB), without regard as to whether prior store data has been globally observed, thereby increasing the throughput of store data and the performance of store operations within the microprocessor.

[0017] Figure 2 illustrates a computer system that may be used in conjunction with at least one embodiment of the invention. A processor 205 accesses data from a cache memory 210 and main memory 215. Illustrated within the processor of Figure 2 is the location of one embodiment of the invention 206. However, embodiments of the invention may be implemented within other devices within the system, as a separate bus agent, or distributed throughout the system. The main memory may be dynamic random-access memory (DRAM), a hard disk drive (HDD) 220, or a memory source 230 located remotely from the computer system containing various storage devices and technologies. The cache memory may be located either within the

processor or in close proximity to the processor, such as on the processor's local bus 207. Furthermore, the cache memory may be composed of relatively fast memory cells, such as six-transistor (6T) cells, or other memory cells of approximately equal or faster access speed.

[0018] Figure 3 illustrates a bus agent in which at least one embodiment of the invention may be used. Particularly, Figure 3 illustrates a microprocessor 301 that contains one or more portions of at least one embodiment of the invention 305. Further illustrated within the microprocessor of Figure 3 is an execution unit 310 to perform operations, such as store operations, within the microprocessor. The exact or relative location of the execution unit and portions of embodiments of the invention are not intended to be limited to those illustrated within Figure 3.

[0019] Figure 4 illustrates one embodiment of the invention in which a GoSB 401 is used to track store operations and store corresponding data values that have become globally observable. Each entry 405 of the GOSB in Figure 2 contains an index value field 406 with which the entry can be referenced, an address value field 407 to indicate target address of the store operation, a data value field 408 to store the data associated with the store operation, a counter field 409 to count a number of store operations that have yet to become globally observable and a valid bit field 410 to indicate whether the data corresponding to the globally observable store operation is available and stored in the data field of the GoSB.

[0020] Also illustrated in the Figure 4 is a non-committed store queue (NcSQ) 415. The NcSQ stores data and address information corresponding to store operations that have been stored in either the line-fill buffer (LFB) 420 or the level-1 (L1) cache

425, but have yet become globally observable. In the embodiment illustrated in Figure 4, the NcSQ is a first-in-first-out (FIFO) queue that has entries containing an address field 416 to store address information corresponding to a particular store operation, a data field 417 to store data corresponding to the store operation, and a GoSB index field 418 to store index information to reference the corresponding entry within the GoSB.

[0021] In the embodiment illustrated in Figure 4, store operations are issued, transferred, or read from the store buffer 430 and stored within NcSQ and either the L1 cache or the LFB and a corresponding entry is allocated within the GoSB. After the store data becomes globally observable, the data is stored into the corresponding GoSB entry from NcSQ. .

[0022] As store data corresponding to a particular target address are stored in the NcSQ, the corresponding counter field in the GoSB is incremented. As store operations become globally observable, the corresponding store address and data are removed from the NcSQ and the corresponding counter field within the GoSB is decremented. After a GoSB counter field reaches zero, the corresponding GoSB entry can be de-allocated and reallocated to a new store operation.

[0023] In the embodiment of the invention illustrated in Figure 4, the L1 cache and LFB may each be snooped by one or more bus agents, such as a microprocessor, for a store data. Within each entry of the L1 cache and the LFB is a GoSB index field 426 and a GoSB valid field 427. The GoSB index field indicates to a snooping agent the location of the corresponding store data within the GoSB. The GoSB valid field indicates whether the corresponding GoSB index is valid and whether it has yet to be

globally observable. The GoSB may also be snooped by a bus agent for the data and will provide the data or, alternatively, point to the most valid data to be used by the snooping agent. If both the L1 cache or the LFB and the GoSB contain the requested data, the GoSB provides the data to the requesting agent.

[0024] Figure 5 illustrates an embodiment of the invention in which the GoSB index and GoSB valid fields are not stored within the L1 cache or LFB entries, but instead, the GoSB index 501 field is stored within entries of the store buffer. In the embodiment illustrated in Figure 5, a GoSB entry may be allocated for a store operation as soon as the store operation becomes non-speculative, or “senior”, rather than waiting until the store operation is, read, transferred, or issued from the store buffer to the LFB or L1 cache.

[0025] Alternatively, the GoSB index field 501 may be logically associated with the store buffer and not physically within the same structure as the store buffer by using logic to point to a particular GoSB index field when a corresponding store buffer field is accessed. In either case, the GoSB index field associated with each entry of the store buffer allows snooping agents to locate the store data within the GoSB early so that the snooping agent may retrieve the data as soon as it becomes globally observable within the GoSB. In the embodiment illustrated in Figure 5, read-for-ownership (RFO) operations may be issued before the corresponding store data is stored within the store buffer. Other aspects of the embodiment illustrated in Figure 5 are similar to those already discussed with regard to the embodiment of the invention illustrated in Figure 4.

[0026] Figure 6 is a flow chart illustrating operations associated with at least one embodiment of the invention. Referring to Figure 6, a first store operation is issued from microprocessor execution logic and the corresponding data is stored within a store buffer entry at operation 601. Either before or after the first store operation is issued from the store buffer, a GoSB entry is allocated and a RFO operation is performed to obtain exclusive ownership of a line in the GoSB and either the L1 cache or the LFB at operation 602. The first store operation data is then stored within the NcSQ and either an LFB or an L1 cache entry. The corresponding GoSB's counter is incremented at operation 603.

[0027] A second store operation is issued and the corresponding data stored within a store buffer entry at operation 604. Either before or after the second store operation is issued from the store buffer, a GoSB entry is allocated and an RFO operation is performed to obtain exclusive ownership of a line in GoSB and either the L1 cache or LFB at operation 605. The second store operation is then moved to NcSQ and either LFB or L1 cache, and the corresponding GoSB's counter is incremented at operation 606.

[0028] In at least one embodiment of the invention, the first and second store operation data resides within the LFB and L1 cache within the same period of time. If the RFO data corresponding to the second store is returned from the L1 cache or LFB prior to the first store operation's data being globally observable, the second store operation is merged into the appropriate entry of the L1 and/or LFB, but not into the corresponding entry of the GoSB, at operation 607. However, if the first store operation's data is globally observable before the second store operation's RFO data

is returned from the L1 cache or LFB, the second store operation data may be merged into the appropriate entry of the GoSB at operation 608. A counter is either incremented or decremented to indicate the number of data associated with a particular store operation allocated within the GoSB that have not, or have, become globally observable, respectively.

[0029] Any or all portions of the embodiments of the invention illustrated herein may be implemented in a number of ways, including, but not limited to, logic using complimentary metal-oxide-semiconductor (CMOS) circuit devices (hardware), instructions stored within a storage medium (software), which when executed by a machine, such as a microprocessor, cause the microprocessor to perform operations described herein, or a combination of hardware and software. References to “microprocessor” or “processor” made herein are intended to refer to any machine or device that is capable of performing operations as a result of receiving one or more input signals or instructions, including CMOS devices.

[0030] Although the invention has been described with reference to illustrative embodiments, this description is not intended to be construed in a limiting sense. Various modifications of the illustrative embodiments, as well as other embodiments, which are apparent to persons skilled in the art to which the invention pertains are deemed to lie within the spirit and scope of the invention.